NUTTX 기반 드론 비행조종컴퓨터의 통합시험을 위한 프로세서 모니터링 연구

[1*]최진원

Development of Processor Real-Time Monitoring Software for Drone Flight Control Computer Based on NUTTX

[1*]Choi Jinwon

요 약

드론과 무인항공기에 탑재되는 비행제어시스템은 설계단계에서부터 철저한 검증이 필수적이며, 이러한 검증은 비행제어 통합시험환경을 통해 이루어진다. 일반적으로 비행제어컴퓨터의 내부 상태를 실시간으로 모니터링하기 위해서는 별도의 디버거를 이용한다. 실시간 메모리 참조 및 Trace가 가능한 Emulator는 비교적 고가이고, JTAG Emulator은 실시간 동작이 불가능 하거나 현재의 고속 프로세서의 처리속도를 따라잡을 수 없는 한계가 있다. 본 논문에서는 NUTTX 기반 드론 비행조종컴퓨터 프로세서의 내부 모니터링 소프트웨어를 개발한 결과를 기술하였으며, 기능시험을 통해 그 기능이 정상적으로 동작되는 것을 확인할 수 있었다. 본 연구 결과는 상용 Debugger와 비교하여 제공되는 기능은 제한적이지만, 예산이 제한적인 상황에서 본 시스템을 활용하여 비행제어시스템 검증에 충분히 사용할 수 있을 것으로 판단된다.

Abstract

Flight control systems installed on unmanned aircraft require thorough verification from the design stage. This verification is made through the integrated flight control test environment. Typically, a debugger is used to monitor the internal state of a flight control computer in real time. Emulator with a real-time memory monitor and trace is relatively expensive. The JTAG Emulator is unable to operate in real time and has limitations that cannot be caught up with the processing speed of latest high-speed processors. In this paper, we describe the results of the development of internal monitoring software for drone flight control computer processors based on NUTTX/PIXHAWK. The results of this study show that the functions provided compared to commercial debugger are limited, but it can be sufficiently used to verify the flight control system using this system under limited budget.

Keywords: AVIONICS, HILS, SIL, JTAG, DEBUGGER

1 한화시스템 수석연구원 (koma.jw@gmail.com)*

Received: Aug. 31, 2022, Revised: Oct. 21, 2022, Accepted: Oct. 30, 2022

I. 서론

4 차산업혁명의 핵심 기술 중 하나인 드론은 세계 주요국가들의 주요 역점 기술이며 드론 산업은 ‘26 년까지 연평균 29%의 급성장이 예상되며 글로벌 항공산업 전체에서도 가장 빠른 성장이 전망되고 있다. 특히 멀티콥터 형태의 드론은 단순한 구조와 제어 원리 그리고 저가형 비행 제어 시스템 보급으로 민간분야에서 많은 관심이 증가되고 있다[1]. 민간 드론 산업이 급속도로 발전한 배경에는 오픈소스 기반 생태계(플랫폼) 조성이 있다. 그 중 비행제어 시스템(Pixhawk), 지상체(QGroundControl), 데이터링크(MAVLink)로 구성되는 PX4 플랫폼은 다른 오픈소스와 달리 상업적으로 사용하여도 공개할 의무가 없으며 또한 누구나 코드를 받아 자유롭게 수정할 수 있는 장점이 있어, 국내외 많은 사용자들이 사용하고 있다[2].

드론 산업이 실생활에 밀접하게 적용되고 사용자 수도 증가하는 만큼, 안전사고도 빈번히 일어나면서 안전 운용에 대한 요구가 높아지고 있다. 드론 산업이 기하급수적으로 증대됨에 따라 드론산업에 대한 전반적인 법적/제도적 개선 필요성에 대한 검토가 이루어지고 있으며, 최근 방위사업청은 군이 운용하는 소형 회전익 무인기 시스템의 비행안정성을 보장하기 위해 ‘소형 회전익 무인기 시스템 감항인증기준(안)’을 마련했다고 밝혔다.

드론을 포함한 무인항공기의 최초안전비행 감항성 입증을 위해서는 주요 항목에 대한 검증을 비행시험 전에 수행해야 하며 비행제어시스템 에서는 HILS(Hardware in the Loop Simulation)를 구축하여 설계단계에서부터 검증하는 것이 필수적이다. 비행제어시스템 및 비행운용프로그램이 요구사항을 충족함을 확인하기 위해서는 비행제어컴퓨터에 대한 독립개채 검증 및 확인시험, 그리고 결함모드 검증 및 확인시험을 수행한다[3]. 상기와 같은 검증 및 확인 시험을 수행하기 위해서는 비행운용프로그램이 탑재된 비행제어컴퓨터의 외부 입출력 신호를 실시간으로 모사하여 운용하면서 소프트웨어 요구도 검증을 수행할 수 있어야 하며, 장비의 입출력신호 및 소프트웨어 내부 변수의 상태를 실시간으로 모니터링하기 위한 자동화된 검증환경이 요구된다[4]. 소프트웨어 요구도 검증을 위해서는 프로세서의 내부 모니터링 및 제어가 필요하며 JTAG Debugger, ICD(In Circuit Debugger)를 비행제어 컴퓨터 외부에 장착하는 방식을 일반적으로 사용한다. ARM 계열의 ETM(Embedded Trace Macrocell), PowerPC 계열의 NEXUS 를 이용한 ICD 장비는 비교적 고가이고, JTAG Debugger 는 실시간 동작이 제한적이고 현재의 고속 프로세서의 처리속도를 따라잡을 수 없는 한계가 있다[5]. 본 연구에서는 PIXHAWK 프로세서 내부 디버깅 레지스터를 이용하여 디버그 에이전트 방식의 프로세서 모니터링 소프트웨어를 개발하고, 해당 소프트웨어 활용을 위한 통합시험환경을 설계한 결과를 기술하였다.

II. 프로세서 모니터링 시스템 구성

2.1 비행제어 컴퓨터와 비행제어 운용 프로그램

비행제어 컴퓨터는 비행체를 안정적으로 제어하여 비행 및 임무를 수행할 수 있는 장치이며, 탑재되는 비행제어 운용 프로그램(OFP: Operational Flight Program)은 고신뢰성이 요구되는 비행안전 필수(Flight Safety Critical) 소프트웨어이다. 전세계 많은 드론 유저들은 아두이노 기반의 오픈소스 프로젝트인 픽스호크(또는 PX4) 하드웨어와 소프트웨어를 사용한다. 픽스호크의 비행제어 컴퓨터는 일반적으로 32 비트 ARM 아키텍처인 STM32 계열 MCU 를 주 CPU 로 사용한다. 본 연구에서는 STM32 계열 CPU 중 STM32F103 보드를 이용하여 OFP 를 구현하였다.

2.2 실시간 운영체제 NUTTX

PX4 의 비행제어 운용 프로그램은 NUTTX 라는 실시간 운영체제에서 동작하며, 본 연구에서 구현한 OFP 역시 NUTTX 운영체제에서 구현되었다. NUTTX 는 임베디드 시스템의 플랫폼으로 ARM, AVR, AVR32, HCS12, LM32, MIPS, RISC-V, SuperH, Xtensa XL6, Z80 등에 사용가능한 운영 체제이다. 커널 유형은 마이크로커널(Microkernel)이다. NuttX 는 리소스를 최소한으로 구성하여 최적의 성능을 얻기 위한 저 수준에서 매우 효율적이며 주요 특징은 다음과 같다[6].

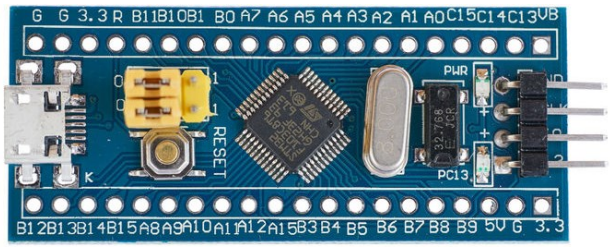

Figure 1. STM32 Flight Control Computer
그림 1. STM32 기반 비행조종컴퓨터

- 표준을 준수하는 C 로 쓰여진 커널 (GNU/Linux 빌드방식)
- 모듈형 설계
- BSD 소켓 인터페이스
- 대칭 멀티 프로세싱 (Symmetric Multi-Processing, SMP)
- 쓰레드 로컬 저장소 (TLS,Thread Local Storage)
- 네트워크 파일 시스템(NFS)
- 유닉스및 다양한 표준 네트워크 프로토콜 지원
- Nutt Shell 제공

2.4 시험 호스트

시험 호스트는 본 연구에서 개발한 프로세서 모니터링 소프트웨어를 원격제어 하여 시험을 수행하고 결과를 확인하는 시험 호스트이다. 시험 호스트는 RS-232 통신을 통해 모니터링 및 시험 입력을 명령하고 수행결과를 화면에 도시하거나 파일로 저장하게 된다. 본 연구에서는 파이썬 언어를 기반으로 한 스크립트로 운용되는 호스트를 구현한다.

2.5 개발 환경

본 연구에서 구현된 디버그 에이전트의 기능 확인을 위한 개발환경을 구성하였으며, 주요 사양은 다음과 같다.

Table 1. Test Environment Specification
표 1 시험 및 운용 환경 사양

명칭	사양
Target Processor	ARM Cortex-M3 STM32F103C8T6
Target OS	Nuttx v10.0.0
Host	Ubuntu 18.04
Compiler	gcc-arm-none-eabi-5_4-2016q2
계측기	Agilent DSOX2012A Oscilloscope

III. 프로세서 모니터링 소프트웨어 개발

3.1 개요

드론 및 무인항공기의 비행제어시스템과 소프트웨어 요구도 검증을 위해서는 프로세서의 내부 모니터링 및 제어가 필요하며, 이 기능은 비행제어컴퓨터와 비행운용프로그램에 영향을 주지 않는 Non-Intrusive Debugging 이 가능해야 한다. 일반적으로 프로세서 모니터링을 위해 JTAG Debugger, ICD Debugger 등의 부가적인 디버깅 장비를 비행조종 컴퓨터의 Debugging Interface 와 연결하여 사용한다. 하지만 JTAG Debugger 는 실시간 동작이 제한적이고, ICD Debugger 는 비교적 고가의 장비로서 소형 드론 개발과 같이 예산이 제한적일수 있는 상황에서 해당 장비의도입은

금전적인 부담이 될 수 있다. 해외 선진항공사의 비행제어 팀은 이러한 제약사항을 해결하기 위해 비행제어 전용 RTOS 를 개발하고 Kernel Mode 디버깅이 가능한 디버그 에이전트를 함께 개발하여 비행제어컴퓨터 실시간 모니터링에 사용하고 있다[1].본 연구에서는 별도의 장비 없이 비행제어 컴퓨터에 탑재되어 프로세서 내부 디버깅 레지스터를 이용하는 디버그 에이전트 방식의 프로세서 모니터링 소프트웨어를 구현하고, 서버 프로그램 개발을 통해 프로세서 내부 모니터링 및 OFP 디버깅 기능을 구현하였으며 소프트웨어 구성 및 운용개념은 다음과 같다.

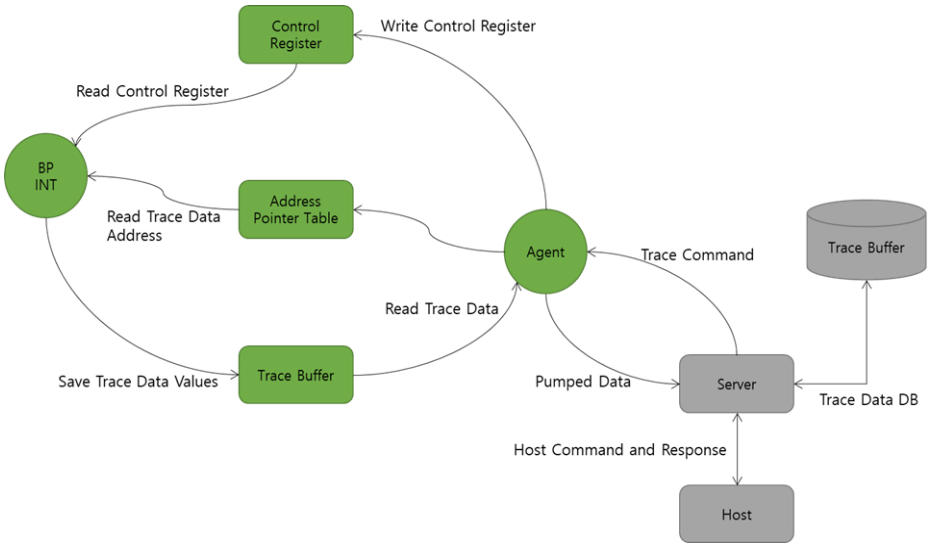

Figure 2. Operational Concept of Debug Agent Software
그림 2. 프로세서 모니터링 소프트웨어 운용개념

3.2 디버그 에이전트

디버그 에이전트는 OFP 내부에 탑재되고 비행제어 컴퓨터의 프로세서가 제공하는 디버깅 레지스터를 활용하여 내부 모니터링 및 디버깅에 필요한 주요 기능을 제공한다. 본 연구에 사용된 ARM 계열 STM32 프로세서는 디버깅을 위한 다양한 레지스터와 인터럽트 벡터 테이블을 제공한다[7].

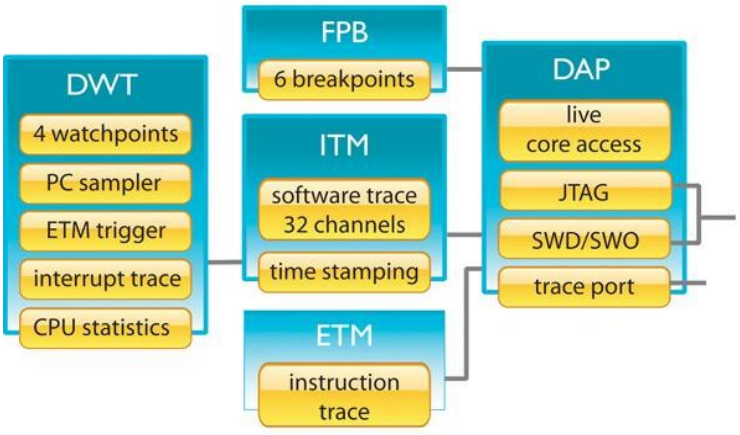

Figure 3. The Debug Module on Cortex-M3/M4[8]
그림 3. Coretex-M3/M4 의 디버그 모듈[8]

디버그 에이전트에 필요한 레지스터는 DHCSR, DEMCR, DFSR, FPB 이며, 그에 대한 설명은 다음과 같다.

Table 2. Debug Registers
표 2. 디버그 레지스터

약어	명칭	내용
DHCSR	Debug Halting Control and Status Register	Debugging 기능 활성화 여부를 제어
DEMCR	Debug Exception and Monitor Control Register	Single Step 기능 활성화 여부를 제어
DFSR	Debug Fault Status Register	Debug Exception 발생 시 Breakpoint 활성화 여부를 확인
FPB	Flash Patch and Breakpoint Unit	Hardware Breakpoint 주소를 설정

또한 디버그 에이전트는 OFP 기능 수행 후 프로세서 IDLE TIME 에 동작하여 탑재 소프트웨어의 영향을 최소화하도록 구현하였으며 동작개념과 타이밍은 다음과 같다.

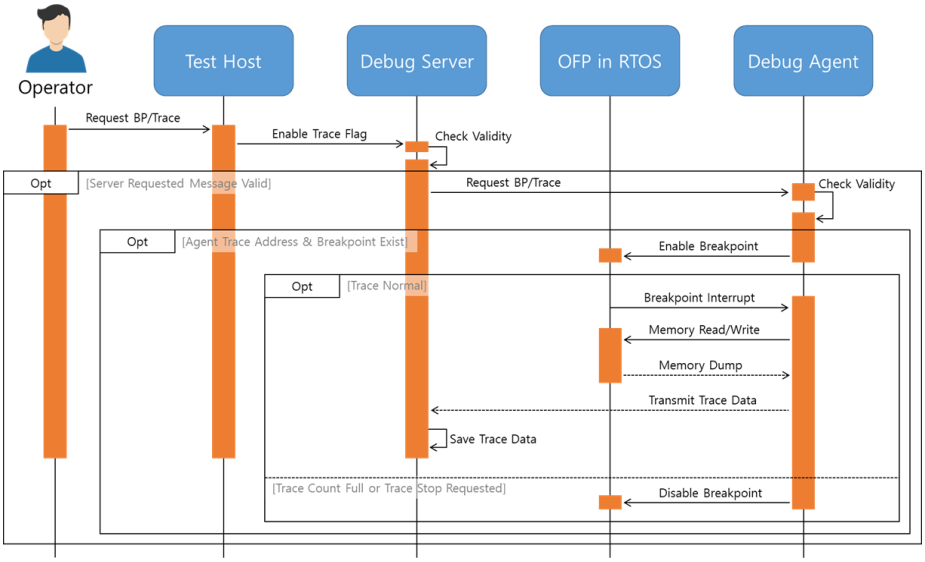

Figure 4. Debug Agent Scenario
그림 4. 디버그 에이전트 시퀀스

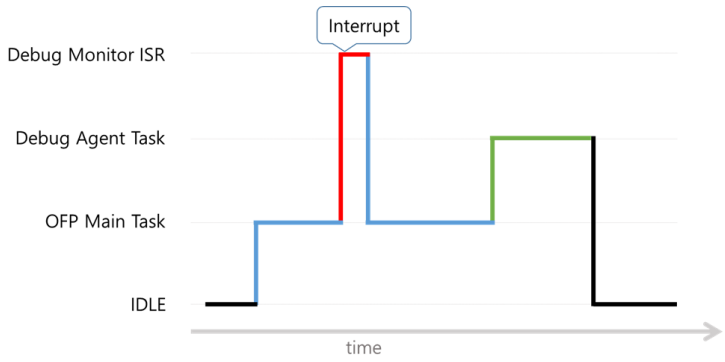

Figure 5. OFP with Debug Agent Task
그림 5. 디버그 에이전트 동작 타이밍

3.3 디버깅 서버

프로세서 모니터링 서버는 디버그 에이전트와 Serial 연동을 통해 통합시험환경 호스트로부터 전달되는 다양한 모니터링 및 디버그 명령을 수행한다. 프로세서 모니터링 서버는 데스크탑 환경

에서 상용 OS 및 데이터베이스로 구성되며, 제한된 자원조건에서 운용되는 디버그 에이전트의 기능을 보조한다.

3.4 시험 호스트

시험 호스트는 스크립트 기반으로 디버깅 서버를 제어하며 디버깅 서버와 TCP/IP 연동을 통해 다음과 같은 기능을 제공한다.

- Memory Read/Write
- Register Read/Write
- Hardware Breakpoint
- Memory Cycle
- Watchdog Timer En/Disable
- Memory Snapshot

3.5 시험 및 결과

시험 절차는 Python 언어 기반 호스트 스크립트로 ICD 를 통해 식별된 각 기능을 Shell 을 통해 확인하였으며, 인터페이스 항목은 다음과 같다.

Table 3. Command List
표 3. 연동 항목

From	To	Description
Host	Agent	Breakpoint Setup
Host	Agent	Memory Read/Write
Host	Agent	Register Read/Write
Host	Agent	Memory Snapshot Request
Host	Agent	Memory Cyclic Write Request
Host	Agent	Memory BP Read/Write Request
Host	Agent	Watchdog Timer En/Disable
Agent	Host	Memory Snapshot Results
Agent	Host	Memory BP Read Request Results
Agent	Host	Memory Cyclic Write Results
Agent	Host	Memory BP Write Request Results

아래 그림은 OFP Main Task 에서 정기적으로 실행되는 Instruction 에 Breakpoint 설정 하였을 때 다음과 같은 Shell Log 가 출력되고 정상적으로 인터럽트 핸들러가 동작함을 확인할 수 있다.

본 연구를 통해 개발된 프로세서 모니터링 소프트웨어는 표 3.에 명시된 인터페이스 항목에 대한 기능시험을 수행하였으며 그 기능이 정상적으로 동작되는 것을 확인할 수 있었다. 다만, 본 연구에서는 비교적 간단한 연산을 수행하는 가상의 OFP 에서 시험을 수행하였으며, 실제 비행제어 컴퓨터와 OFP 가 탑재된 환경에서 탑재 소프트웨어 동작에 영향을 주지 않는지 추후 검증이 필요하다.

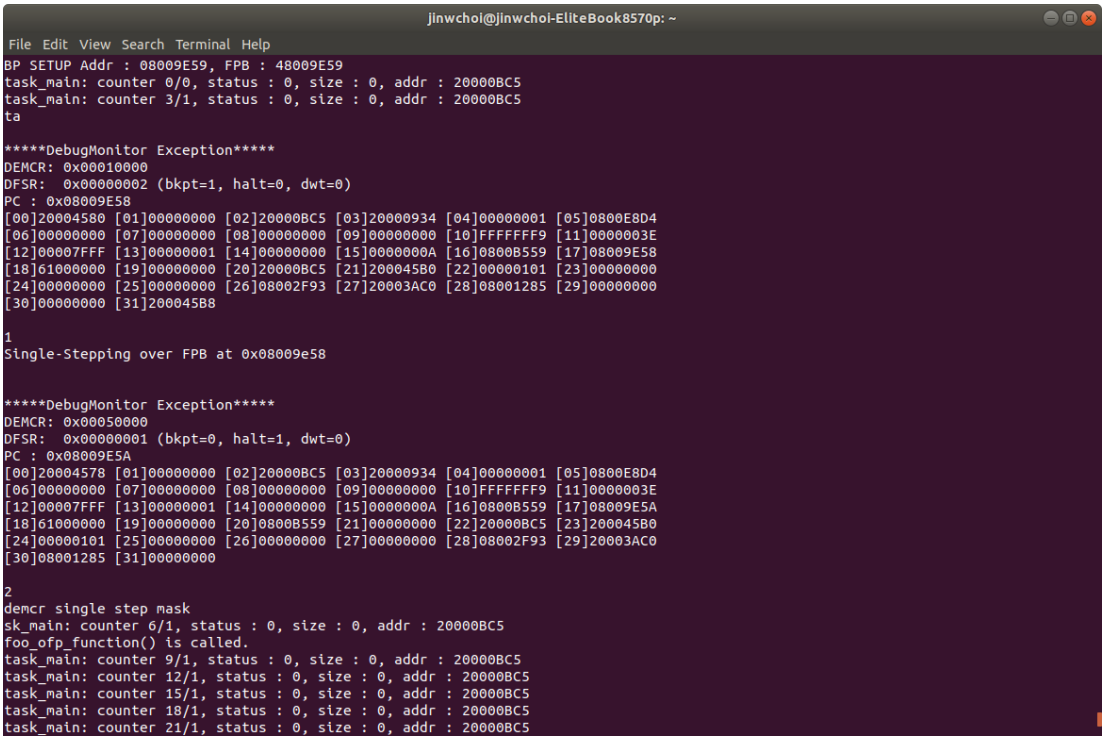

Figure 6. Breakpoint Shell Log
그림 6. BP 쉘 로그 화면

IV. 결론

본 논문에서는 드론 비행제어시스템의 OFP 내부 모니터링 및 디버깅을 위한 에이전트 기반의 프로세서 모니터링 소프트웨어를 개발하였다. PX4 플랫폼에서 일반적으로 사용하는 ARM STM32 프로세서에 NUTTX 실시간 운영체제에서 해당 기능을 구현하였으며, 상용 Debugger 와 비교하여 제공되는 기능은 제한적이지만, 예산이 제한적인 상황에서 본 시스템을 활용하여 비행제어시스템 검증에 충분히 사용할 수 있을 것으로 판단된다. 본 연구에서는 프로세서모니터링 소프트웨어를 구현하고 단위 기능에 대한 검증을 수행하였으나, 추후 시험환경을 구현하고 PIXHAWK Apps 와 디버그 에이전트 통합을 통해 실제 ICD Debugger 와 비교하여 프로세서 모니터링 장비로서 해당 소프트웨어의 사용 가능성에 대한 연구를 수행할 예정이다. 이외에도 디버그 에이전트를 진단 기능으로 확장하여 Serial 통신 기반 텔레메트리 연동을 통해 비행조종 컴퓨터의 원격 정비가 가능하도록 확장할 예정이다.

V. 참고문헌

[1] Chang, T. J., "Regulatory environment and structural change of UAV industry," Journal of Aerospace System Engineering, Vol. 9, No. 3, pp.17~22, 2015.
[2] PX4 (2021, Feb.). Px4 Pro Autopliot Software. Github. [Online]. Available: https://github.com/PX4/Firmware/
[3] Choi, S. K., and Moon, J. H., “Airworthiness Case Study for the Tactical UAV's Flight Control System,” Journal of the KIMST, Vol. 17, No. 4, pp.430-435, 2014.
[4] Choi, J. W., and Kang, B. R., “Development of integrated real-time monitoring system for UAV flight control system,” The society for Aerospace System Engineering Fall Conference, 2017, pp.7~8.
[5] Lee, C., and Kim, J. C., “A Study on Processor Monitoring for Integration Test of Flight Control

Computer equipped with A Modern Processor," Journal of Institute of Control, Robotics and Systems, vol. 14, no. 10, pp. 395-399, Oct. 2008.
[6] NuttX (2021, Feb.). NuttX. Wikipedia. [Online]. Available: https://ko.wikipedia.org/wiki/NuttX
[7] ARM, "ARM v7-M Architecture Reference Manual," December 2014.
[8] Lotta Frimanson and Anders Lundgren, "Lotta Frimanson and Anders Lundgren," Convergence Promotions LLC, April 2011.

저자소개

최진원 (*Choi Jinwon*)

2003 년 2 월 한국항공대학교 항공기계공학 학사
2005 년 2 월 한국항공대학교 항공우주 및 기계공학 석사
2005 년 3 월~2012 년 2 월 썬에어로시스 기술연구소 과장
2012 년 3 월~2021 년 3 월 대한항공 항공기술연구원 과장
2021 년 4 월~현재 한화시스템 항공개발센터 수석연구원

관심분야: HILS, SIL 비행제어, 항공전자